unnecessarily search for and find hidden functions, thereby preventing the user from being distracted from the task at hand. Moreover, the GUIs described and claimed herein allow software packages to be used to their fullest capabilities and also widen the spectrum of users that will be able to utilize these capabilities since even a relatively inexperienced GUI operator will have the ability to gain access to all of the software's functionality very easily. These results have not heretofore been achieved in the art.

[0012] These and other features, objects and benefits of the invention will be best understood by those skilled in the art by reading the following detailed description of the invention in conjunction with the drawings which are first described briefly below.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] In the drawings wherein like reference numerals refer to like elements throughout the several views hereof:

[0014] Figure 1 is a view of a prior art GUI having various option areas which may be activated by a user to invoke software functionality;

[0015] Figure 2 is a view of the GUI of Figure 2 wherein the option areas have been re-sized;

[0016] Figure 3 is a flow chart of a preferred method of the present invention wherein hidden options may be accessed by a user;

[0017] Figure 4 is a view of a GUI provided in accordance with the present invention;

[0018] Figure 5 is a view of the GUI of Figure 4 having a separate menu that has been accessed by a user and which displays software options; and

[0018] Figure 6 is an alternative embodiment of a GUI of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0019] Referring now to Figure 3, a flow chart of preferred methods for providing GUIs in accordance with the present invention is illustrated. It will be appreciated by those with skill in the art that the inventive GUIs will be programmed in an appropriate software language and run on a computer or server. The GUIs may be written in any software language such as C++, JAVA, Visual Basic, HTML, UNIX, or other, preferably object-oriented, language. The GUIs will be manipulatable by a user of the software using an input device, such as a mouse, touch sensitive pad, or other mechanism.

[0020] The preferred methods begin at step 60, and at step 70 a user attempts to access the inventive GUI for invocation of software options that the user desires to use. It is then determined at step 80 whether the options are hidden from the user on the GUI. If the options are not hidden, then the method progresses to step 90 and the user accesses the desired option so that the particular software functionality can be invoked. Alternatively, even if all of the menu or option areas are visible, it may still be desirable to put up the menu of Figure 5 since it may be poorly designed or otherwise hidden by another element on the screen, for example. The method then stops at step 100.

[0021] However, if the options are indeed hidden from the user, or the user cannot otherwise find the particular option that it desires to invoke, or if it is otherwise desirable to up the menu, then the method progresses to step 110 wherein the user accesses an instrumentality of the GUI (to be described in more detail below) to display at least a partial list of the options invocable for the software, some of which may have been hidden from the user in the GUI for one reason or another. The hidden options may be displayed to the user after the user accesses the GUI instrumentality in a variety of ways. For example, but not intending to limit the invention in any way, a separate menu of all of the options available to the user, hidden or not, may be displayed to the user after the instrumentality is invoked so that the user may then see all of the options on the menu and pick the desired option therefrom. Alternatively, an option tree structure in the form of bullets may appear, or a folder/subfolder arrangement may be displayed having all of

the options thereon for display to the user. All such arrangements and equivalents thereof are intended to be within the scope of the present invention.

[0022] Regardless of how the hidden options are displayed to the user, at step 120 the user selects from the displayed list of hidden options in the GUI, the desired option which the user wishes to invoke. This software function is then conventionally performed, and at step 140 the user determines whether there are other software options which it desires to be invoked. If not, then the method stops at step 100. If so, then the method returns to step 80 where it is determined whether the options are hidden, and the process repeats.

[0023] In this fashion, the GUIs of the present invention always make readily available to the user all of the allowed software options that a user may desire to invoke when operating the GUI and the software. This allows the users fast access to the options without having to first search throughout all of the option areas of a GUI for the desired option or options to be invoked. The inventive GUIs thereby allow a user to operate the software very quickly and efficiently while alleviating any frustration that a user my experience searching for options. These results have not heretofore been achieved in the art.

[0024] Figure 4 illustrates an inventive GUI which is similar in form to the prior GUIs of Figures 1 and 3 in that the same three option areas 30, 40, 50 are extant. In accordance with the invention, an instrumentality 150 is provided to the GUI which will allow the user to display the options which are available in any one, or all, or the option areas 30, 40 50, preferably by a single mouse click. In the preferred embodiment, instrumentality 150 is an icon in the form of a lightning bolt. It will be appreciated that the icon could take any form and may be placed in the GUI at any place that is conspicuous to the user. Alternatively, a unique combination or sequence of keystrokes could be used as the instrumentality, for example, Ctrl-Shift-F1.